

AMENDMENTS TO THE CLAIMS

1. (Currently amended) A method for updating peer entities in a communication network comprising:

among a first quantity of update messages, formatting one or more of the update messages to establish a first quantity of formatted update messages for a first peer entity,

wherein said first peer entity is one of a peer group and a peer,

wherein the first quantity of the formatted update messages is less than or equal to the first quantity of update messages; and

among the formatted update messages, enqueueing to a queue, one or more formatted update messages, to establish a quantity of enqueued formatted update messages, wherein the quantity of enqueued formatted update messages is less or equal to the quantity of formatted update messages, and

wherein at least one of the quantity of formatted update messages and the quantity of the enqueued formatted update messages is a programmable quantity;

after all of the first quantity of update messages are formatted in said step of formatting messages for said first peer entity, formatting a second quantity of update messages for a second peer entity

wherein the quantity of update messages is a first quantity of 1 to m update messages, wherein said step of formatting the quantity of update messages to be formatted comprises formatting 1 to n update messages among the quantity of update messages to establish a first quantity of formatted update messages, wherein n is less than m, wherein the steps of the method are performed in respect of each peer entity and all of the first quantity of formatted update messages are enqueued and advertised to their respective peer entities;

maintaining a point of reference to a data structure in which one of the nth update message and the n + 1 update message is stored;

after all of the 1 to n update messages have been formatted and advertised to each peer entity, formatting one or more update messages comprising the n + 1 update message to establish a second quantity of formatted update messages, wherein the data structure is identified with said point of reference.

2. (Original) The method recited in claim 1, further comprising storing the formatted update messages in a cache associated with the peer entity.

3. (Original) The method recited in claim 1, further comprising replicating one of the formatted update messages to establish a replica formatted update message.

4. (Original) The method recited in claim 3, wherein said peer entity is a peer group, further comprising enqueueing said replica formatted update message to a queue associated with a peer in said peer group.

5. (Original) The method recited in claim 4, further comprising transmitting a replica formatted update message from the queue associated with the peer group to a peer in the peer group.

6. (Original) The method recited in claim 1, wherein each of the update messages is formatted.

7. (Original) The method recited in claim 1, wherein each of the formatted update messages is enqueued.

8. (Original) The method recited in claim 6, wherein each of the quantity of formatted update messages is enqueued.

9. (Previously presented) The method recited in claim 1, wherein the quantity of update messages to be formatted is programmable.

10. (Previously presented) The method recited in claim 1, wherein the quantity of formatted update messages to be enqueued is programmable, the method further comprising:  
programmatically receiving the quantity of formatted update messages to be enqueued;  
after all of the formatted update messages are enqueued in said step of enqueueing messages for said first peer entity, formatting a second quantity of update messages for a second peer entity.

11. (Canceled)

12. (Original) The method recited in claim 1, wherein the quantity of update messages to be formatted is unequal to the quantity of formatted update messages to be enqueued.

13. (Original) The method recited in claim 1, wherein the quantity of update messages to be formatted is programmable, the method further comprising:  
programmatically receiving the quantity of update messages to be formatted; and  
after the update messages of the programmed quantity are formatted in said step of formatting, transmitting enqueued messages from the queue.

14. (Original) The method recited in claim 1, wherein the quantity of formatted update messages to be enqueued is programmable, the method further comprising:  
programming the quantity of formatted update messages to be enqueued; and  
after the formatted update messages of the programmed quantity are enqueued in said step of enqueueing, transmitting enqueued messages from the queue.

15. (Original) The method recited in claim 1, wherein the quantity of update messages to be formatted is programmable, the method further comprising:  
if an amount of available memory is less than a threshold amount, ignoring the programmable quantity of quantity of update messages to be formatted; and  
formatting a pre-determined number of update messages.

16. (Original) The method recited in claim 1, wherein the quantity of formatted update messages to be enqueued is programmable, the method further comprising:  
if an amount of available memory is less than a threshold amount, ignoring the programmable quantity of messages to be enqueued; and  
enqueueing a pre-determined number of messages.

17. (Original) The method recited in claim 1, wherein said method is performed by a border gateway protocol process or application of a network device operating system.

18. (Original) The method recited in claim 1, wherein the quantity of update messages is a first quantity of update messages and the quantity of update messages to be formatted is programmable, said method further comprising:  
programmatically receiving the quantity of formatted update messages; and

after formatting said quantity of the first quantity of update messages and after enqueueing said quantity of formatted update messages, performing the steps of claim 1 on a second quantity of update messages for the first peer entity.

19. (Original) The method recited in claim 18, wherein the step of formatting the second quantity of update messages comprises formatting a predetermined quantity of update messages, and ignoring the programmed quantity of messages to be formatted.

20. (Original) The method recited in claim 1, wherein the step of enqueueing comprises a step of suspending enqueueing of formatted update messages, if a memory limit is reached during formatting of update messages.

21. (Canceled)

22. (Currently amended) The method recited in claim-24\_1, wherein said point of reference is a pointer that points to said data structure.

23.-31. (Canceled)

32. (Currently amended) A computer-readable volatile or non-volatile storage medium comprising storing one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out ~~the steps recited in claim 1;~~  
among a first quantity of update messages, formatting one or more of the update messages to establish a first quantity of formatted update messages for a first peer entity,  
wherein said first peer entity is one of a peer group and a peer,

wherein the first quantity of the formatted update messages is less than or equal to the first quantity of update messages; and

among the formatted update messages, enqueueing to a queue, one or more formatted update messages, to establish a quantity of enqueued formatted update messages, wherein the quantity of enqueued formatted update messages is less or equal to the quantity of formatted update messages, and

wherein at least one of the quantity of formatted update messages and the quantity of the enqueued formatted update messages is a programmable quantity;

after all of the first quantity of update messages are formatted in said step of formatting messages for said first peer entity, formatting a second quantity of update messages for a second peer entity

wherein the quantity of update messages is a first quantity of 1 to m update messages, wherein said step of formatting the quantity of update messages to be formatted comprises formatting 1 to n update messages among the quantity of update messages to establish a first quantity of formatted update messages, wherein n is less than m, wherein the steps of the method are performed in respect of each peer entity and all of the first quantity of formatted update messages are enqueued and advertised to their respective peer entities;

maintaining a point of reference to a data structure in which one of the nth update message and the n + 1 update message is stored;

after all of the 1 to n update messages have been formatted and advertised to each peer entity, formatting one or more update messages comprising the n + 1 update message to establish a second quantity of formatted update messages, wherein the data structure is identified with said point of reference.

33. (Currently amended) ~~A~~The computer-readable medium of claim 32 further comprising one or more sequences of instructions, which when executed by one or more

processors, cause the one or more processors to carry out ~~the steps recited in claim 2~~ storing the formatted update messages in a cache associated with the peer entity.

34. (Currently amended) ~~A~~ The computer-readable medium of claim 32 further comprising ~~one or more~~ sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out ~~the steps recited in claim 3~~ replicating one of the formatted update messages to establish a replica formatted update message.

35. (Currently amended) ~~A~~ The computer-readable medium of claim 32 wherein said peer entity is a peer group and further comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out ~~the steps recited in claim 4~~ enqueueing said replica formatted update message to a queue associated with a peer in said peer group.

36. (Currently amended) ~~A~~ The computer-readable medium of claim 32 further comprising ~~one or more~~ sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out ~~the steps recited in claim 5~~ transmitting a replica formatted update message from the queue associated with the peer group to a peer in the peer group.

37. (Currently amended) ~~A~~ The computer-readable medium ~~comprising one or more~~ sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out ~~the steps recited in claim 6~~ of claim 32 wherein each of the update messages is formatted.

38. (Currently amended) ~~A-The computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 7 of claim 32 wherein each of the formatted update messages is enqueued.~~

39. (Currently amended) ~~A-The computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 8 of claim 32 wherein each of the quantity of formatted update messages is enqueued.~~

40. (Currently amended) ~~A-The computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 9 of claim 32 wherein the quantity of update messages to be formatted is programmable.~~

41. (Currently amended) ~~A-The computer-readable medium of claim 32 wherein the quantity of formatted update messages to be enqueued is programmable and further comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 10 programmatically receiving the quantity of formatted update messages to be enqueued; after all of the formatted update messages are enqueued in said step of enqueueing messages for said first peer entity, formatting a second quantity of update messages for a second peer entity.~~

42. (Canceled)



43. (Currently amended) ~~A-The computer-readable medium of claim 32 comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 12 wherein the quantity of update messages to be formatted is unequal to the quantity of formatted update messages to be enqueued.~~

44. (Currently amended) ~~A-The computer-readable medium of claim 32 wherein the quantity of update messages to be formatted is programmable and further comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 13 programmatically receiving the quantity of update messages to be formatted; and after the update messages of the programmed quantity are formatted in said step of formatting, transmitting enqueued messages from the queue.~~

45. (Currently amended) ~~A-The computer-readable medium of claim 32 wherein the quantity of formatted update messages to be enqueued is programmable and further comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 14 programming the quantity of formatted update messages to be enqueued; and after the formatted update messages of the programmed quantity are enqueued in said step of enqueueing, transmitting enqueued messages from the queue.~~

46. (Currently amended) ~~A-The computer-readable medium of claim 32 wherein the quantity of update messages to be formatted is programmable and further comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 15; if an amount of available memory is~~

less than a threshold amount, ignoring the programmable quantity of quantity of update messages to be formatted; and formatting a pre-determined number of update messages.

47. (Currently amended) ~~A-~~The computer-readable medium of claim 32 wherein the quantity of formatted update messages to be enqueued is programmable and further comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 16: if an amount of available memory is less than a threshold amount, ignoring the programmable quantity of messages to be enqueued; and enqueueing a pre-determined number of messages.

48. (Canceled)

49. (Currently amended) ~~A-~~The computer-readable medium of claim 32 wherein the quantity of update messages is a first quantity of update messages and the quantity of update messages to be formatted is programmable and further comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 18programmatically receiving the quantity of formatted update messages; and after formatting said quantity of the first quantity of update messages and after enqueueing said quantity of formatted update messages, performing the steps of claim 1 on a second quantity of update messages for the first peer entity.

50. (Currently amended) ~~A-~~The computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 19of claim 32 wherein the instructions that cause formatting the second quantity of update messages comprise instructions that cause

formatting a predetermined quantity of update messages, and ignoring the programmed quantity of messages to be formatted.

51. (Currently amended) ~~A~~The computer-readable medium comprising one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to carry out the steps recited in claim 20 of claim 32 wherein the instructions that cause enqueueing comprise instructions that cause suspending enqueueing of formatted update messages, if a memory limit is reached during formatting of update messages.

52.-62. (Canceled)

63. (New) A router, comprising:  
one or more processors;  
one or more network interfaces coupled to the one or more processors  
a storage device coupled to the one or more processors and storing one or more sequences of instructions which when executed by the one or more processors cause the one or more processors to update peer entities in a communication network by performing:  
among a first quantity of update messages, formatting one or more of the update messages to establish a first quantity of formatted update messages for a first peer entity, wherein said first peer entity is one of a peer group and a peer,  
wherein the first quantity of the formatted update messages is less than or equal to the first quantity of update messages; and  
among the formatted update messages, enqueueing to a queue, one or more formatted update messages, to establish a quantity of enqueued formatted update messages, wherein the quantity of enqueued formatted update messages is less or equal to the quantity of formatted update messages, and

wherein at least one of the quantity of formatted update messages and the quantity of the enqueued formatted update messages is a programmable quantity;

after all of the first quantity of update messages are formatted in said step of formatting messages for said first peer entity, formatting a second quantity of update messages for a second peer entity

wherein the quantity of update messages is a first quantity of 1 to m update messages, wherein said step of formatting the quantity of update messages to be formatted comprises formatting 1 to n update messages among the quantity of update messages to establish a first quantity of formatted update messages, wherein n is less than m, wherein the steps of the method are performed in respect of each peer entity and all of the first quantity of formatted update messages are enqueued and advertised to their respective peer entities;

maintaining a point of reference to a data structure in which one of the nth update message and the n + 1 update message is stored;

after all of the 1 to n update messages have been formatted and advertised to each peer entity, formatting one or more update messages comprising the n + 1 update message to establish a second quantity of formatted update messages, wherein the data structure is identified with said point of reference.

64. (New) The router recited in claim 63, further comprising instructions which when executed cause storing the formatted update messages in a cache associated with the peer entity.

65. (New) The router recited in claim 63, further comprising instructions which when executed cause replicating one of the formatted update messages to establish a replica formatted update message.

66. (New) The router recited in claim 63, further comprising instructions which when executed cause each of the update messages to be formatted.

67. (New) The router recited in claim 63, further comprising instructions which when executed cause each of the formatted update messages to be enqueued.

68. (New) The router recited in claim 63, wherein the quantity of update messages to be formatted is programmable.

69. (New) The router recited in claim 63, wherein the quantity of formatted update messages to be enqueued is programmable, the router further comprising instructions which when executed cause programmatically receiving the quantity of formatted update messages to be enqueued; after all of the formatted update messages are enqueued in said step of enqueueing messages for said first peer entity, formatting a second quantity of update messages for a second peer entity.

70. (New) The router recited in claim 63, wherein the quantity of update messages to be formatted is unequal to the quantity of formatted update messages to be enqueued.

71. (New) The router recited in claim 63, wherein the quantity of update messages to be formatted is programmable, the router further comprising instructions which when executed cause programmatically receiving the quantity of update messages to be formatted; and after the update messages of the programmed quantity are formatted in said step of formatting, transmitting enqueued messages from the queue.

72. (New) The router recited in claim 63, wherein the quantity of formatted update messages to be enqueued is programmable, the router further comprising instructions which when executed cause programming the quantity of formatted update messages to be enqueued; and after the formatted update messages of the programmed quantity are enqueued in said step of enqueueing, transmitting enqueued messages from the queue.

73. (New) The router recited in claim 63, wherein the quantity of update messages to be formatted is programmable, the router further comprising instructions which when executed cause: if an amount of available memory is less than a threshold amount, ignoring the programmable quantity of quantity of update messages to be formatted; and formatting a pre-determined number of update messages.

74. (New) The router recited in claim 63, wherein the quantity of formatted update messages to be enqueued is programmable, the router further comprising instructions which when executed cause: if an amount of available memory is less than a threshold amount, ignoring the programmable quantity of messages to be enqueued; and enqueueing a pre-determined number of messages.

75. (New) The router recited in claim 63, wherein the recited steps are performed by a border gateway protocol process or application of a network device operating system.

76. (New) The router recited in claim 63, wherein the quantity of update messages is a first quantity of update messages and the quantity of update messages to be formatted is programmable, said router further comprising instructions which when executed cause programmatically receiving the quantity of formatted update messages; and after formatting said quantity of the first quantity of update messages and after enqueueing said quantity of formatted

update messages, performing the steps of claim 63 on a second quantity of update messages for the first peer entity.

77. (New) The router recited in claim 63, wherein the step of enqueueing comprises a step of suspending enqueueing of formatted update messages, if a memory limit is reached during formatting of update messages.

78. (New) The router recited in claim 63, wherein the quantity of update messages is a first quantity of 1 to m update messages, wherein said step of formatting the quantity of update messages to be formatted comprises formatting 1 to n update messages among the quantity of update messages to establish a first quantity of formatted update messages, wherein n is less than m, wherein the steps of the router recited in claim 63 are performed in respect of each peer entity and all of the first quantity of formatted update messages are enqueued and advertised to their respective peer entities, the router further comprising instructions which when executed cause maintaining a point of reference to a data structure in which one of the nth update message and the n + 1 update message is stored; after all of the 1 to n update messages have been formatted and advertised to each peer entity, formatting one or more update messages comprising the n + 1 update message to establish a second quantity of formatted update messages, wherein the data structure is identified with said point of reference.